(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 January 2019 (17.01.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/011636 A1

(51) International Patent Classification:
H04N 19/96 (2014.01) *G06T 17/00* (2006.01) *G06T 9/40* (2006.01) *G06T 17/10* (2006.01) *H04N 19/147* (2014.01) *H04N 19/20* (2014.01) *H04N 19/597* (2014.01)

(21) International Application Number: PCT/EP2018/066923

(22) International Filing Date: 25 June 2018 (25.06.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
17305931.2 13 July 2017 (13.07.2017) EP

(71) Applicant: **INTERDIGITAL VC HOLDINGS, INC.** [US/US]; 1209 Orange Street, Wilmington, DE 19801 (US).

(72) Inventors: **LASSERRE, Sebastien**; c/o TECHNICOLOR, 975 Avenue des Champs-Blancs, CS17616, 35576 CESSON-SEVIGNE (FR). **RICARD, Julien**; c/o TECHNICOLOR, 975 Avenue des Champs-Blancs, CS17616, 35576 CESSON-SEVIGNE (FR). **GUEDE, Celine**; c/o TECHNICOLOR, 975 Avenue des Champs-Blancs, CS17616, 35576 CESSON-SEVIGNE (FR).

(74) Agent: **LABELLE, Lilian** et al.; TECHNICOLOR, 1-5, Rue Jeanne d'Arc, 92130 Issy-les-Moulineaux (FR).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

(54) Title: A METHOD AND APPARATUS FOR ENCODING/DECODING THE GEOMETRY OF A POINT CLOUD REPRESENTING A 3D OBJECT

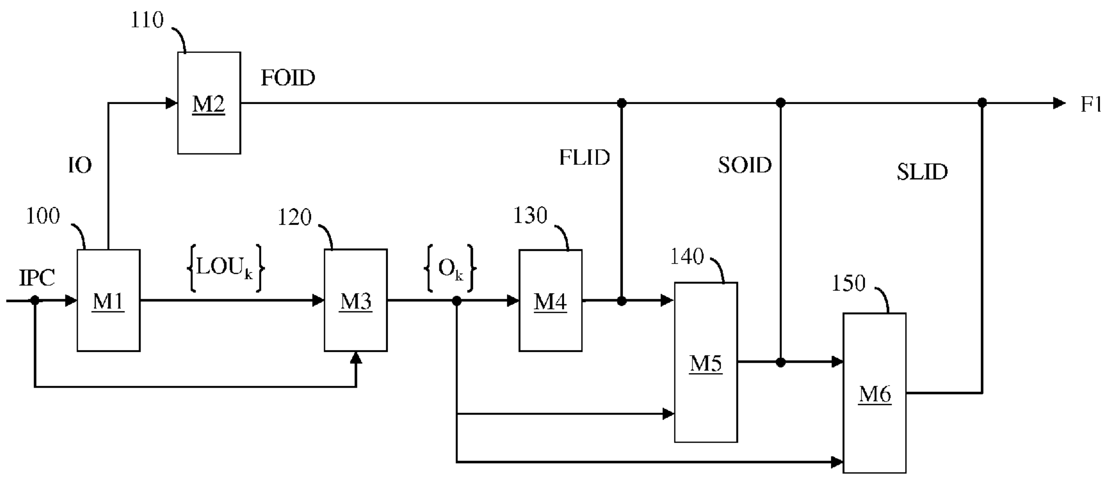


Fig. 1

(57) Abstract: The present principles relates to a two-steps approach for encoding and decoding the geometry of a point cloud. In a first step, an octree-based structure is obtained by splitting recursively a cube encompassing the point cloud until the leaf cubes associated with the leaf nodes of said octree-based structure reach down an expected size. In a second step, for each leaf cube associated with the leaf nodes of said octree-based structure (IO), the approach determines if a local octree-based structure is associated (or not) with a leaf cube by using a Rate-Distorsion Optimisation process that optimizes a trade-off between a bit-rate for encoding a candidate octree-based structure approximating the geometry of points of the point cloud which are included in said leaf cube of the octree-based structure, and a distortion that takes into account spatial distances between, on one hand, said points of the point cloud, and on the other hand, points included in leaf cubes associated with leaf nodes of the candidate octree-based structure.

WO 2019/011636 A1

[Continued on next page]

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

1. Title

A method and apparatus for encoding/decoding the geometry of a point cloud representing a 3D object.

2. Field.

The present principles generally relate to coding and decoding of a point cloud representing the geometry of a point cloud representing a 3D object. Particularly, but not exclusively, the technical field of the present principles are related to octree-based encoding/decoding of point cloud.

3. Background.

The present section is intended to introduce the reader to various aspects of art, which may be related to various aspects of the present principles that are described and/or claimed below. This discussion is believed to be helpful in providing the reader with background information to facilitate a better understanding of the various aspects of the present principles. Accordingly, it should be understood that these statements are to be read in this light, and not as admissions of prior art.

A point cloud is a set of points usually intended to represent the external surface of a 3D object but also more complex geometries like hair, fur that may not be represented efficiently by other data format like meshes. Each point of a point cloud is often defined by a 3D spatial location (*X*, *Y*, and *Z* coordinates in the 3D space) and possibly by other associated attributes such as color, represented in the RGB or YUV color space for example, a transparency, a reflectance, a two-component normal vector, etc.

A colored point cloud may be a set of 6-components points (X, Y, Z, R, G, B) or equivalently (X, Y, Z, Y, U, V) where (X,Y,Z) defines the spatial location of a point in a 3D space and (R,G,B) or (Y,U,V) defines a color of this point.

In the following the term “point cloud” refers to any point cloud including a colored point cloud.

Colored point clouds may be static or dynamic depending on whether or not the cloud evolves with respect to time. It should be noticed that in case of a dynamic point cloud, the number of points is not constant but, on the contrary, generally evolves with time. A dynamic point cloud is thus a time-ordered list of sets of points.

Practically, colored point clouds may be used for various purposes such as culture heritage/buildings in which objects like statues or buildings are scanned in 3D in order to share the spatial configuration of the object without sending or visiting it. Also, it is a way to ensure preserving the knowledge of the object in case it may be destroyed; for instance, a temple by an earthquake. Such colored point clouds are typically static and huge.

Another use case is in topography and cartography in which, by using 3D representations, maps are not limited to the plane and may include the relief.

Automotive industry and autonomous cars are also domains in which point clouds may be used. Autonomous cars should be able to “probe” their environment to take safe driving decision based on the reality of their immediate neighboring. Typical sensors produce dynamic point clouds that are used by the decision engine. These point clouds are not intended to be viewed by a human being. They are typically small, not necessarily colored, and dynamic with a high frequency of capture. They may have other attributes like the reflectance that is a valuable information correlated to the material of the physical surface of sensed object and may help the decision.

Virtual Reality (VR) and immersive worlds have become a hot topic recently and foreseen by many as the future of 2D flat video. The basic idea is to immerse the viewer in an environment all round him by opposition to standard TV where he can only look at the virtual world in front of him. There are several gradations in the immersivity depending on the freedom of the viewer in the environment. Colored point clouds are a good format candidate to distribute VR worlds. They may be static or dynamic and are typically of averaged size, say no more than a few millions of points at a time.

Point cloud compression will succeed in storing/transmitting 3D objects for immersive worlds only if the size of the bitstream is low enough to allow a practical storage/transmission to the end-user.

It is also crucial to be able to distribute dynamic point clouds to the end-user with a reasonable consumption of bandwidth while maintaining an acceptable (or preferably very good) quality of experience. Similarly to video compression, a good use of temporal correlation is thought to be the crucial element that will lead to efficient compression of dynamic point clouds.

Well-known approaches project a colored point cloud representing the geometry and colors of a 3D object, onto the faces of a cube encompassing the 3D object to obtain videos on texture and depth, and code the texture and depth videos using a legacy encoder such as 3D-HEVC (an extension of HEVC whose specification is found at the ITU website, T recommendation, H series, h265, http://www.itu.int/rec/T-REC-H.265-201612-I/en annex G and I).

Performance compression is close to video compression for each projected point, but some contents may be more complex because of occlusions, redundancy and temporal stability when dynamic point clouds are considered. Consequently, point cloud compression is more demanding than video compression in term of bit-rates.

Regarding occlusions, it is virtually impossible to get the full geometry of a complex topology without using many projections. The required resources (computing power, storage memory) for encoding/decoding all these projections are thus usually too high.

Octree-based encoding is also a well-known approach for encoding the geometry of a point cloud. An octree-based structure is obtained for representing the geometry of the point cloud by splitting recursively a cube encompassing the point cloud until the leaf cubes, associated with the leaf nodes of said octree-based structure, contain no more than one point of the point cloud. The spatial locations of the leaf nodes of the octree-based structure thus represent the spatial locations of the points of the point cloud, i.e. its geometry.

Such splitting process thus requires important resources in term of computing power because the splitting decision are done over the whole point cloud which may comprise a huge number of points.

Therefore, there is a trade-off to be found between obtaining a good representation of the geometry of a point cloud without using an optimization process having a high computing complexity.

4. Summary.

The following presents a simplified summary of the present principles to provide a basic understanding of some aspects of the present principles. This summary is not an extensive overview of the present principles. It is not intended to identify key or critical elements of the present principles. The following summary merely presents some aspects of the present principles in a simplified form as a prelude to the more detailed description provided below.

Generally speaking, the present principles relates to a two-steps approach for encoding the geometry of a point cloud. In a first step, an octree-based structure is obtained by splitting recursively a cube encompassing the point cloud until the leaf cubes associated with the leaf nodes of said octree-based structure reach down an expected size. In a second step, for each leaf cube associated with the leaf nodes of said octree-based structure (IO), the approach determines if a local octree-based structure is associated (or not) with a leaf cube by using a Rate-Distorsion Optimisation process that optimizes a trade-off between a bit-rate for encoding a candidate octree-based structure approximating the geometry of points of the point cloud which are included in said leaf cube of the octree-based structure, and a distortion that takes into account spatial distances between, on one hand, said points of the point cloud, and on the other hand, points included in leaf cubes associated with leaf nodes of the candidate octree-based structure.

Representing the geometry of a point cloud by the octree-based structure (step 1) and local octree-based structures (step 2) is advantageous because it allows to determine locally an optimal representation of the geometry, i.e. the optimization process optimizes the octree-based on a smaller amount of points, thus reducing dramatically the complexity of optimization which is usually done over the whole set of points of the point cloud.

Another advantage is to profit from the possibility of prediction of a local octree-based structure by an already coded neighboring octree-based structure. This advantage is similar to the advantage of decomposing an image into coding blocks as performed in many video compression standards, for instance in HEVC, and then using intra prediction between blocks (here intra prediction of octree-based structure).

Also, considering, dynamic point clouds, it is possible to obtain a temporal prediction of a local octree-based structure from already coded points at a preceding time. Again, this advantage is similar to the advantage of inter temporal prediction between blocks as applied in many video compression standards. Using local octree-based structures allows for a practical motion search because it is performed on a reasonable amount of points.

The present principles relate a method and a device. The method comprises:

- determining if points of a point cloud included in a leaf cube associated with a leaf node of an octree-based structure representing the geometry of the point cloud are approximated by at least one candidate point of a candidate octree-based structure, as a function of by a bit-rate for encoding said candidate octree-based structure and a distortion obtained from spatial distances between, on one hand, said points of the point clouds and on the other hand, said candidate points, a leaf cube associated with a leaf node of said candidate octree-based structure including at most a single candidate point located in its center;

- encoding a first leaf node information data indicating if a candidate octree-based structure has been determined for approximating said points of the point cloud; and

- if said points of the point cloud are approximated by at least one candidate point, encoding said candidate octree-based structure.

According to an embodiment, encoding said candidate octree-based structure comprises:

- encoding, an octree information data representative of said candidate octree-based structure; and

- encoding a second leaf node information data indicating if a leaf cube associated with a leaf node of said candidate octree-based structure includes a single candidate point located in its center.

According to an embodiment, the method further obtaining said octree-based structure by splitting recursively a cube encompassing the point cloud until the leaf cubes, associated with the leaf nodes of said octree-based structure, reach down an expected size.

According to an embodiment, the method further comprises encoding or the processor of the device is further configured to encode another octree information data representative of the octree-based structure.

According to an embodiment, the distortion is obtained from spatial distances between, on one hand, said points of the point cloud, and on the other hand, said candidate points together with points of an inverse-projected point cloud which are included in said leaf cube associated with a leaf node of said octree-based structure representing the geometry of a point cloud, said inverse-projected point cloud being obtained by inverse-projecting at least one depth image representative of at least one point of the point cloud.

According to an embodiment, the distortion is also obtained from neighboring points which are included in at least one neighboring cube of said leaf cube associated with a leaf node of said octree-based structure representing the geometry of a point cloud, a neighboring point being either a point of said inverse-projected point cloud which is included in said at least one neighboring cube or a candidate point located in the center of a leaf cube associated with a leaf node of a candidate octree-based structure previously determined for one of said at least one neighboring cube.

According to an embodiment, a candidate octree-based structure is obtained by pruning an initial octree-based structure at a given level (le).

According to another of their aspects, the present principles relate to another method and device. Said other method comprises:

- if a first leaf node information data indicates that a local octree-based structure has been determined for approximating points of a point cloud included in a leaf cube associated with a leaf node of an octree-based structure representing the geometry of said point cloud,

  - obtaining a local octree-based structure for said leaf cube from an octree information data;

  - if a second leaf node information data indicates that a leaf cube associated with a leaf node of said local octree-based structure includes a single point, adding a single point in the center of said leaf cube.

According to an embodiment, the method further comprises:

- obtaining an octree-based structure from another octree information data;

- for at least one current leaf node of said octree-based structure,

  - if a first leaf node information data indicates that a local octree-based structure has been determined for approximating points of a point cloud included in a leaf cube associated with said current leaf node,

  - obtaining a local octree-based structure for said leaf cube from an octree information data;

  - if a second leaf node information data indicates that a leaf cube associated with a leaf node of said local octree-based structure includes a single point, adding a single point in the center of said leaf cube.

According to another of their aspects, the present principles relate to a signal carrying:

- a first octree information data representative of an octree-based structure representing the geometry of a point cloud;

- for at least one current leaf cube associated with a current leaf node of said octree-based structure,

  - a first leaf node information data indicating if a local octree-based structure has been determined for said current leaf cube;

  - if said first leaf node information data indicates that a local octree-based structure has been determined

    - a second octree information data representative of said determined local octree-based structure; and

    - at least one second leaf node information data indicating if a leaf cube associated with a leaf node of said local octree-based structure includes a single point in its center.

According to another of their aspects, the present principles relate to a non-transitory computer readable medium comprising instructions which, when executed by a computer, cause the computer to carry out one of the above method.

According to another of their aspects, the present principles relate to a computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out the steps of one of the above methods.

The specific nature of the present principles as well as other objects, advantages, features and uses of the present principles will become evident from the following description of examples taken in conjunction with the accompanying drawings.

5. Brief Description of Drawings.

In the drawings, examples of the present principles are illustrated. It shows:

- **Fig. 1** shows schematically a diagram of the steps of the method for encoding the geometry of a point cloud representing a 3D object in accordance with an example of the present principles;

- **Fig. 2** illustrates an example of an octree-based structure;

- **Fig. 3** shows the diagram of the sub-steps of the step 120 in accordance with an embodiment of the present principles;

- **Fig. 4** shows an illustration of an example of a candidate octree-based structure;

- **Fig. 5** shows an illustration of an example of neighboring Largest Octree Units;

- **Fig. 6** shows schematically a diagram of the method for pruning an initial candidate octree-based structure at a given level in accordance with an example of the present principles;

- **Fig. 7a-b** illustrate examples of pruning an octree-based structure at level 3;

- **Fig. 8a-b** illustrate examples of pruning an octree-based structure at level 2;

- **Fig. 9a-b** illustrate examples of pruning an octree-based structure at level 1;

- **Fig. 10** shows schematically a diagram of the steps of the method for decoding, from a bitstream, the geometry of a point cloud representing a 3D object in accordance with an example of the present principles;

- **Fig. 11** shows an example of an architecture of a device in accordance with an example of present principles; and

- **Fig. 12** shows two remote devices communicating over a communication network in accordance with an example of present principles;

- **Fig. 13** shows the syntax of a signal in accordance with an example of present principles.

Similar or same elements are referenced with the same reference numbers.

6. Description of Example of the present principles.

The present principles will be described more fully hereinafter with reference to the accompanying figures, in which examples of the present principles are shown. The present principles may, however, be embodied in many alternate forms and should not be construed as limited to the examples set forth herein. Accordingly, while the present principles are susceptible to various modifications and alternative forms, specific examples thereof are shown by way of examples in the drawings and will herein be described in detail. It should be understood, however, that there is no intent to limit the present principles to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the present principles as defined by the claims.

The terminology used herein is for the purpose of describing particular examples only and is not intended to be limiting of the present principles. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises", "comprising," "includes" and/or "including" when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof. Moreover, when an element is referred to as being "responsive" or "connected" to another element, it can be directly responsive or connected to the other element, or intervening elements may be present. In contrast, when an element is referred to as being "directly responsive" or "directly connected" to other element, there are no intervening elements present. As used herein the term "and/or" includes any and all combinations of one or more of the associated listed items and may be abbreviated as"/".

It will be understood that, although the terms first, second, etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and, similarly, a second element could be termed a first element without departing from the teachings of the present principles.

Although some of the diagrams include arrows on communication paths to show a primary direction of communication, it is to be understood that communication may occur in the opposite direction to the depicted arrows.

Some examples are described with regard to block diagrams and operational flowcharts in which each block represents a circuit element, module, or portion of code which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that in other implementations, the function(s) noted in the blocks may occur out of the order noted. For example, two blocks shown in succession may, in fact, be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending on the functionality involved.

Reference herein to “in accordance with an example” or “in an example” means that a particular feature, structure, or characteristic described in connection with the example can be included in at least one implementation of the present principles. The appearances of the phrase in accordance with an example” or “in an example” in various places in the specification are not necessarily all referring to the same example, nor are separate or alternative examples necessarily mutually exclusive of other examples.

Reference numerals appearing in the claims are by way of illustration only and shall have no limiting effect on the scope of the claims.

While not explicitly described, the present examples and variants may be employed in any combination or sub-combination.

The present principles are described for encoding/decoding a colored point cloud but extends to the encoding/decoding of a sequence of colored point clouds because each colored point cloud of the sequence is sequentially encoded/decoded as described below.

In the following, an image contains one or several arrays of samples (pixel values) in a specific image/video format which specifies all information relative to the pixel values of an image (or a video) and all information which may be used by a display and/or any other device to visualize and/or decode an image (or video) for example. An image comprises at least one component, in the shape of a first array of samples, usually a luma (or luminance) component, and, possibly, at least one other component, in the shape of at least one other array of samples, usually a color component. Or, equivalently, the same information may also be represented by a set of arrays of color samples, such as the traditional tri-chromatic RGB representation.

A pixel value is represented by a vector of **nv** values, where **nv** is the number of components. Each value of a vector is represented with a number of bits which defines a maximal dynamic range of the pixel values.

A depth image is an image whose pixel values depths of 3D points. Usually, a depth image is a grey levels image.

An octree-based structure comprises a root node, at least one leaf node and possibly intermediate nodes. A leaf node is a node of the octree-based cube which has no child. All other nodes have children. Each node of an octree-based structure is associated with a cube. Thus, an octree-based structure comprises a set $\{C_j\}$ of at least one cube C_j associated with node(s).

A leaf cube is a cube associated with a leaf node of an octree-based structure.

In the example illustrated on **Fig. 2**, the cube associated with the root node (depth 0) is split into 8 sub-cubes (depth 1) and two sub-cubes of depth 1 are then split into 8 sub-cubes (last depth = maximum depth = 2).

The sizes of the cubes of a same depth are usually the same but the present principles are not limited to this example. A specific process may also determine different numbers of sub-cubes per depth, when a cube is split, and/or multiple sizes of cubes of a same depth or according to their depths.

In the following, the term “local octree-based structure determined for a cube” refers to an octree-based structure determined in the 3D space delimited by the cube that encompasses a part of the point cloud to be encoded.

In the opposite, a global octree-based structure refers to an octree-based structure determined in a 3D space delimited by the cube that encompasses the point cloud to be encoded.

Fig. 1 shows schematically a diagram of the steps of the method for encoding the geometry of a point cloud IPC representing a 3D object in accordance with an example of the present principles.

In step 100, a module M1 determines an octree-based structure IO comprising at least one cube, by splitting recursively a cube encompassing the point cloud until the leaf cubes, associated with the leaf nodes of said octree-based structure IO, reach down an expected size.

The leaf cubes associated with the leaf nodes of the octree-based structure IO may then include or not points of the point cloud IPC. A leaf cube associated with a leaf node of the octree-based structure IO is named in the following a Largest Octree Unit (LOU_k), **k** means an index referencing the Largest Octree Unit associated with a leaf node **k** of the octree-based structure IO.

In step 110, a module M2 encodes a first octree information data FOID representative of the octree-based structure IO.

In step 120, for each LOU_k, a module M3 determines if a local octree-based structure O_k is associated with a LOU_k by optimizing a trade-off between a bit-rate $R_{k,n}$ for encoding a candidate octree-based structure $O_{k,n}$ approximating the geometry of points $P_{k,or}$ of the point cloud IPC which are included in said LOU_k, and a distortion $D_{k,n}$ taking into account spatial distances between, on one hand, said points $P_{k,or}$ of the point cloud IPC, and on the other hand, points $P_{k,n}$ included in leaf cubes associated with leaf nodes of the candidate octree-based structure $O_{k,n}$.

Mathematically speaking, the distortion $D_{k,n}$ is a metric given by:

$$D_{k,n} = d(P_{k,n}, P_{k,OR}) + d(P_{k,OR}, P_{k,n})$$

where d(A,B) is a metric that measures the spatial distance from a set of points A to a set of points B. This metric is not symmetric, this means that distance from A to B differs from the distance from B to A.

The distance $d(P_{k,n}, P_{k,OR})$ ensures that the points included in leaf cubes associated with leaf nodes of a candidate octree-based structure $O_{k,n}$ are not too far from the point cloud IPC, avoiding coding irrelevant points.

The distance $d(P_{k,OR}, P_{k,n})$ ensures that each point of the point cloud IPC is approximated by points not too far from them, i.e. ensures that all parts of the point cloud IPC are well approximated.

According to an embodiment, the distance d(A,B) is given by:

$$d(A,B) = \sum_{p \in A} \|p - q_{\text{closest}}(p,B)\|_2^2$$

where the norm is the Euclidan distance and $q_{\text{closest}}(p,B)$ is the closest point of B from a point p of A defined as

$$q_{\text{closest}}(p,B) = \underset{q \in B}{\operatorname{argmin}} \|p - q\|_2^2.$$

According to the optimization process, it may be happen that a LOU_k does not include any point of the point cloud IPC. In that case the LOU_k is named a non-coded LOU_k.

It may also happen that the optimization process determines that the points of the point cloud IPC which are included in the LOU_k are not represented (coded) by any candidate octree-based structure $O_{k,n}$. This is the case when the cost for coding those points is too high relatively to the cost associated with $R_{k,n} = 0$ and the distortion $D_{k,n}$ obtained between already coded points, from other already coded LOU_k for example, and $P_{k,OR}$.

In step 130, for each LOU_k, a module M4 encodes a first leaf node information data FLID indicating if a local octree-based structure O_k has been determined for said LOU_k.

If a first leaf node information data FLID indicates that a local octree-based structure O_k has been determined for a LOU_k, in step 140 a module M5 encodes a second octree information data SOID representative of said determined local octree-based structure O_k, and in step 150, a module M6 encodes a second leaf node information data SLID indicating if a leaf cube of said local octree-based structure O_k includes a point representative of a part of the point cloud IPC.

The first octree information data FOID, the first leaf node information data FLID, the second octree information data SOID and the second leaf node information data SLID may be stored and/or transmitted in a bitstream F1.

According to an embodiment of step 110 and/or 140, illustrated on **Fig. 2**, the first octree information data FOID data and the second octree information data SOID comprises a binary flag per node which equal to 1 to indicate that a cube associated with said node is split and to 0 otherwise.

According to an embodiment of step 120, the first leaf node information data FLID comprises a binary flag per leaf node which equal to 1 to indicate if a local octree-based structure O_k has been determined for a LOU_k and to 0 otherwise.

According to an optional variant, the module M2 also generates a maximum depth of the cube splitting.

This avoids signaling first octree information data for all cubes having the maximum depth.

According to an embodiment of step 150, the second leaf node information data SLID comprises a binary flag per leaf node which equal to 1 to indicate if a leaf cube of a local octree-based structure O_k includes a point representative of a part of the point cloud IPC and to 0 otherwise.

According to an embodiment, the first octree information data FOID, the first leaf node information data FLID, the second octree information data SOID and/or the second leaf node information data SLID may be coded using an entropy coder like CABAC (a description of the CABAC is found in the specification of HEVC at http://www.itu.int/rec/T-REC-H.265-201612-I/en).

Entropy encoding the second octree information data SOID and/or the second leaf node information data SLID may be efficient in term of coding, because specific contexts may be used to code the binary flags per node because usually only a few nodes of an octree-based structure are split and the probability for the binary flags associated with neighboring nodes to have a same value is high.

Fig. 3 shows the diagram of the sub-steps of the step 120 in accordance with an embodiment of the present principles.

As discussed above, an octree-based structure IO comprising at least one LOU_k is obtained, and a Rate Distortion optimization (RDO) process is used to determine a best local octree-based structure O_k for at least one LOU_k.

A single flag may then be encoded in the bitstream F1 to indicate if a LOU_k includes or not a point of the point cloud IPC.

A RDO process that is performed on a LOU_k may find a best local octree-based structure O_k from N candidate octree-based structures $O_{k,n}$ $(n \in [1; N])$. The basic principle is to test successively each candidate octree-based structure $O_{k,n}$ and for each candidate octree-based structure $O_{k,n}$ to calculate a Lagrangian cost $C_{k,n}$ given by:

$$C_{k,n} = D_{k,n} + \lambda R_{k,n} \qquad (1)$$

where $R_{k,n}$ and $D_{k,n}$ are respectively the bit-rate and distortion detailed above, and λ is a fixed Lagrange parameter that may be fixed for all the candidate octree-based structures $O_{k,n}$.

The best local octree-based structure O_k is then obtained by minimizing the Lagrangian cost $C_{k,n}$:

$$O_k = \underset{O_{k,n}}{\operatorname{argmin}} \, C_{k,n}(O_{k,n}) \qquad (2)$$

High values for the Lagrangian parameter strongly penalize the bit-rate $R_{k,n}$ and lead to a low quality of approximation, while low values for the Lagrangian parameter allow easily high values for $R_{k,n}$ and lead to high quality of approximation. The range of values for lambda depends on the distortion metric, the size of the LOU_k, and most importantly the distance between two adjacent points. Assuming that this distance is unity, typical values for lambda are in the range from a few hundreds, for very poor coding, to a tenth of unity for good coding. These values are indicative and may also depend on the content.

Determining a best local octree-based structure O_k for a LOU_k is now detailed in accordance with an embodiment of the present principles.

In step 300, the module M3 obtains a set of N candidate octree-based structures $O_{k,n}$ for the LOU_k and obtains a set of points $P_{k,n}$ for each candidate octree-based structure $O_{k,n}$. The points $P_{k,n}$ are points which are included in cubes associated with leaf nodes of a candidate octree-based structure $O_{k,n}$.

In step 310, the module M3 obtains the bit-rate $R_{k,n}$ for encoding each candidate octree-based structure $O_{k,n}$.

In step 320, the module M3 obtains points $P_{k,or}$ of the point cloud IPC which are included in the LOU_k.

In step 330, the module M3 obtains a distortion $D_{k,n}$ for each candidate octree-based structure $O_{k,n}$, each distortion $D_{k,n}$ takes into account the spatial distances between, on one hand, the points $P_{k,OR}$, and on the other hand, the points $P_{k,n}$.

In step 340, the module M3 calculates the Lagrangian cost $C_{k,n}$ according to equation (1) for each candidate octree-based structure $O_{k,n}$.

In step 350, the module M3 obtains the best local octree-based structure O_k according to equation (2) once all the candidate octree-based structures $O_{k,n}$ have been considered.

Note that as explained above, it may then consider that the best trade-off for a LOU_k is to not code the points included in it. In that case, no local octree-based structure O_k is determined for this LOU_k.

According to an embodiment of step 300, a candidate octree-based structure $O_{k,n}$ comprises at least one leaf node and the leaf cube associated to a leaf node may (or not) include a single point.

Fig. 4 shows an illustration of an example of a candidate octree-based structure $O_{k,n}$ according to this embodiment. This figure represents an example of a quadtree-based structure that splits a square, but the reader will easily extend it to the 3D case by replacing the square by a cube (LOU_k).

According to this example, the cube is split into 4 sub-cubes C1, C2 C3 and C4 (depth 1). The sub-cube C1 is associated with a leaf node and does not contain any point. The sub-cube C2 is recursively split into 4 sub-cubes (depth 2). The sub-cube C3 is also recursively split and the sub-cube C4 is not split but a point, located in the center of the cube for example, is associated with it,…, etc.

On the right part of **Fig. 4** is shown an illustration of the candidate octree-based structure. A black circle indicates that a node is split. A binary flag is associated with each white circle (leaf node) to indicate if the square (a cube in the 3D case) includes (1) or not (0) a point.

According to this example, a point is located in the center of a cube because it avoids any additional information about the spatial location of that point once the cube is identified in the octree-based structure. But the present principles are not limited to this example and may extend to any other spatial location of a point in a cube.

The present principles are not limited to the candidate octree-based structure illustrated on **Fig. 4** but extend to any other octree-based structure comprising at least one leaf node whose associated leaf cube includes at least one point.

According to an embodiment of the step 310, when the second octree information data SOID comprises a binary flag per node which equal to 1 to indicate that a cube associated with said node is split and to 0 otherwise (embodiment of step 140) and the second leaf node information data SLID comprises a binary flag per leaf node which equal to 1 to indicate if a leaf cube of a local octree-based structure O_k includes a point representative of a part of the point cloud IPC and to 0 otherwise (embodiment of step 150), the bit-rate $R_{k,n}$ is the sum of the numbers of the binary flags comprised in the second octree information data SOID and the second leaf node information data SLID.

According to another variant, a first sequence of the binary flags comprised in the second octree information data SOID and a second sequence of binary flags comprised in the second leaf node information data SLID are formed and entropy-coded and the bit-rate $R_{k,n}$ is then obtained from the entropy-encoded versions of said first and second sequences.

According to an alternative to this embodiment of the step 310, the syntax used to encode a candidate octree-based structure $O_{k,n}$ may comprise an index of a table (Look-Up-Table) that identifies a candidate octree-based structure among a set of candidate octree-based structures determined beforehand, for example by an end-user. This table of candidate octree-based structures is known at the decoder.

According to an embodiment, a set of bits (one or more bytes) are used for encoding said index of a table.

The bit-rate $R_{k,n}$ is thus the bit-rate required for encoding said index.

According to a variant of steps 320 and 330, in step 320, the module M3 also obtains points $P_{k,IP}$ of an inverse-projected point cloud IPPC which are included in the LOU_k. Said inverse-projected point cloud IPPC is obtained by inverse-projecting at least one depth images representative of a part of the point cloud IPC, as proposed, for example, in “Image-Based Surface Compression”, Tilo Ochotta & Dietmar Saupe, September 2008, in Computer Graphics Forum.

In step 330, the module M3 obtains a distortion $D_{k,n}$ that takes into account spatial distances between, on one hand, said points $P_{k,or}$ of the point cloud, and on the other hand, points $P_{k,n}$ included in leaf cubes associated with leaf nodes of the candidate octree-based structure $O_{k,n}$ together with the points $P_{k,IP}$.

Mathematically speaking, the distortion $D_{k,n}$ is a metric given by:

$$D_{k,n} = d(P_{k,n} \cup P_{k,IP}, P_{k,OR}) + d(P_{k,OR}, P_{k,n} \cup P_{k,IP})$$

The distance $d(P_{k,OR}, P_{k,n} \cup P_{k,IP})$ ensures that each point of the point cloud IPC is approximated by points not too far from them, i.e. ensures that parts of the point cloud IPC which are not represented by the inverse-projected point cloud IPPC are well approximated.

According to a variant of the steps 320 and 330, in step 320, the module M3 obtains neighboring point $P_{k,NEI}$ which are either points of the inverse point cloud IPPC which are included in at least one neighboring Largest Coding Unit $LOU_{k,NEI}$ of the LOU_k, or points included in leaf cubes associated with leaf nodes of local octree-based structures previously determined for said at least one neighboring Largest Coding Unit $LOU_{k,NEI}$.

In step 330, the module M3 obtains a distortion that also takes into account the spatial distances between the points $P_{k,or}$ and the neighboring points $P_{k,NEI}$.

Mathematically speaking, the distortion $D_{k,n}$ is a metric given by:

$$D = d(P_{k,n} \cup P_{k,IP} \cup P_{k,NEI}, P_{k,OR}) + d(P_{k,OR}, P_{k,n} \cup P_{k,IP} \cup P_{k,NEI})$$

The distance $d(P_{k,OR}, P_{k,n} \cup P_{k,IP} \cup P_{k,NEI})$ ensures also that each point of the point cloud IPC is approximated by points not too far, including also neighboring points included in neighboring Largest Coding Units $LOU_{k,NEI}$. It is advantageous because it avoids coding too finely points of the point cloud IPC, close to the edge of the neighboring Largest Coding Units $LOU_{k,NEI}$ that could be already well represented by points included in the neighboring Largest Coding Units $LOU_{k,NEI}$. Consequently, this saves bit-rates by coding less points, and with a small impact on the distortion.

According to an embodiment of this variant, illustrated on **Fig. 5**, the Largest Coding Unit $LOU_{k,NEI}$ are defined in order to have at least one vertex, one edge or one face in common with the LOU_k..

Fig. 5 shows an illustration of an example of neighboring Largest Coding Units $LOU_{k,NEI}$. This figure represents an example of a quadtree-based structure relative to the LOU_k and eight neighboring $LOU_{k,1-8}$ of the LOU_k. The points $P_{k,OR}$ are represented by white rectangles. The points $P_{k,IP}$ are represented by black rectangles. The points $P_{k,NEI}$ are represented by black circles. The point $P_{k,n}$ are represented by white circles. It is understood that the 2D description is for illustration only. in 3D, one should consider the 26 neighboring cubes instead of the 8 neighboring squares of the 2D illustration.

According to this example, the points $P_{k,NEI}$ are the points included in four $LOU_{k,1-4}$, i.e. points that are either included in cubes associated with leaf nodes of local octree-based structures associated with these four $LOU_{k,1-4}$ (step 820) and/or points of the inverse-projected colored point cloud IPPC which are included in said $LOU_{k,1-4}$.

According to an embodiment of the method, a candidate octree-based structure $O_{k,n}$ approximating the geometry of points ($P_{k,or}$) of the point cloud which are included in a leaf cube LOU_k is obtained by pruning an initial candidate octree-based structure O_n at a given level **le**.

Fig. 6 shows schematically a diagram of the method for pruning an initial candidate octree-based structure O_n at a given level **le** in accordance with an example of the present principles.

In step 600, a module M7 obtains at least one octree-based structure O_n from the points included in the LOU_k.

An octree-based structure O_n is an octree in which each parent node may comprise at most eight children nodes and in which a cube is associated with each of this node. A root node (depth 0) is the unique node without any parent node and each child node (depth greater than 0) has a single parent node.

An octree-based structure O_n may be obtained by splitting recursively the LOU_k. An octree-based structure comprises thus a set of at least one cube associated with nodes(s).

A stopping condition for recursive splitting may be checked when a maximum depth is reached or when the size of cube, associated with a node, is smaller than a threshold or when the number of points of the points included in the LOU_k does not exceed a minimum number.

Changing at least one of those stopping condition may lead to multiple initial candidate octree-based structures O_n. **N** is the number of initial candidate octree-based structures O_n.

In step 610, a module M8 obtains at least one octree-based structure $O_{n,test}$ by pruning an initial candidate octree-based structure O_n at a given level **le** initialized to 1.

Pruning an initial candidate octree-based structure O_n at a level **le** means determining the $\mathbf{I_n}$ (depending of n) nodes **i** at the level **le** which have children, selecting iteratively each node **i** and, for each node **i,** obtaining an octree-based structure $O_{n,test}$ from the initial candidate octree-based structure O_n by erasing all children (and their potential descendants) of the node **i** from the octree-based structure O_n.

Fig. 7a-b, respectively **Fig. 8a-b** and **Fig. 9a-b,** illustrate examples of pruning an octree-based structure at level 3, respectively 2 and 1, in accordance with an example of the present principles.

A black circle indicates that a node is split (has children). A binary flag is associated with each white circle (leaf node) to indicate if a cube associated with this leaf node includes (1) or not (0) at least one 3D point included in the LOU_k.

On the left of these **Figs.** is shown an example of an octree-based structure before the pruning and on the right the octree-based structure after pruning. The pruning level is illustrated by a vertical line and a binary flag (=1) is assigned to the pruned node (designated by a horizontal arrow).

In step 620, a module M9 calculates a Lagrangian cost $C_{n,test}$ given by:

$$C_{n,test} = D_{n,test} + \lambda R_{n,test}$$

where $R_{n,test}$ is a bit-rate for encoding the octree-based structure $O_{n,test}$ approximating the geometry of the points included in the LOU_k, $D_{n,test}$ is the distortion taking into account spatial distances between the points included in the LOU_k and the points $P_{n,test}$, included in leaf cubes associated with leaf nodes of the octree-based structure $O_{n,test}$ and λ is a fixed Lagrange parameter that may be fixed for all the octree-based structures $O_{n,test}$.

In step 630, a module M10 compares the Lagrangian cost $C_{n,test}$ to a Lagrangian cost C_n associated to the octree-based structure O_n. If the Lagrangian cost $C_{n,test}$ is smaller than the Lagrangian cost C_n, then an optimal octree-based structure $O_{n,opt}$ is set to $O_{n,test}$, otherwise it is set to O_n.

The process is repeated iteratively on all nodes **i** and levels **le** by replacing the octree-based structure O_n by the octree-based structure $O_{n,opt}$ in steps 610, 620 and 630. An optimal octree-based structure $O_{n,opt}$ is thus obtained for each initial candidate octree-based structure O_n.

In step 640, for each initial candidate octree-based structure O_n, a module M11 selects the candidate octree-based structure $O_{k,n}$ as being the optimal octree-based structure $O_{n,opt}$ obtained after all pruning. After a candidate octree-based structure $O_{k,n}$ has been obtained for all **n**, the candidate octree-based structure with the least Lagrangian cost is selected to become the optimized octree-based structure O_k associated with the LOU_k.

Fig. 10 shows schematically a diagram of the steps of the method for decoding, from a bitstream, the geometry of a point cloud representing a 3D object in accordance with an example of the present principles.

In step 1000, a module M12 decodes, from the bitstream F1, the first octree information data FOID.

In step 1010, a module M13 obtains an octree-based structure IO from the first octree information data FOID.

In step 1000, a module M12 decodes, from the bitstream F1, a first leaf node information data FLID, and in step 1020, a module M14 determines if a LOU_k associated with a leaf node of the octree-based structure IO is associated with a local octree-based structure O_k from said first leaf node information data FLID.

If a first leaf node information data FLID indicates that a local octree-based structure O_k has been associated with a LOU_k, In step 1000, a module M12 decodes, from the bitstream F1, a second octree information data SOID and a second leaf node information data SLID, and in step 1030 a module M15 obtains a local octree-based structure O_k for said LOU_k from the second octree information data SOID, and in step 1040, a module M16 determines if a leaf cube of said local octree-based structure O_k includes a point representative of a part of the point cloud IPC from the second leaf node information data SLID. In that case, a point is added to the leaf cube of said local octree-based structure Ok, preferably in its center.

The octree-based structure IO in which the LOU_k are replaced by the local octree-based structure O_k represent the geometry of the point cloud.

According to an embodiment, the first octree information data FOID, the first leaf node information data FLID, the second octree information data SOID and/or the second leaf node information data SLID may be obtained by entropy-decoding the bitstream F1. The entropy-decoding may be compliant with a CABAC-like coding.

The present principles have been described by considering leaf cubes LOU_k associated with the leaf nodes of an octree-based structure IO.

According to an embodiment of the encoding and decoding methods, a cube C encompassing at least one point of the point cloud IPC is considered rather than a LOU_k.

The point cloud IPC is then encoded as follows:

The steps 100 and 110 is cancelled.

In step 120, the module M3 determines if a local octree-based structure O_k is associated with the encompassing cube C including at least one point of the point cloud IPC by optimizing a trade-off between a bit-rate $R_{k,n}$ for encoding a candidate octree-based structure $O_{k,n}$ approximating the geometry of points $P_{k,or}$ of the point cloud which are included in said an encompassing cube C, and a distortion $D_{k,n}$ taking into account spatial distances between, on one hand, said points $P_{k,or}$ of the point cloud which are included in said encompassing cube, and on the other hand, points $P_{k,n}$ included in leaf cubes associated with leaf nodes of the candidate octree-based structure $O_{k,n}$.

In step 130, a first leaf node information data FLID indicating if a local octree-based structure O_k has been determined for said encompassing cube C is encoded.

If said first leaf node information data FLID indicates that a local octree-based structure O_k has been determined for said encompassing cube C, in step 140, an octree information data SOID representative of said determined local octree-based structure O_k is encoded, and in step 150, a second leaf node information data SLID, indicating if a leaf cube of said local octree-based structure O_k includes a point representative of a part of the point cloud, is encoded.

The point cloud IPC is then decoded as follows:

In step 1000, a first leaf node information data FLID, indicating if a local octree-based structure O_k has been determined for an encompassing cube C including at least one point representing at least a part of the point cloud to be decoded, is decoded from a bitstream.

If said first leaf node information data FLID indicates that a local octree-based structure O_k has been associated with said encompassing cube C, in step 1000, an octree information data SOID representative of said determined local octree-based structure O_k and a second leaf node information data SLID indicating if a leaf cube of said local octree-based structure O_k includes a point representative of a part of the point cloud, are decoded from a bitstream. In step 1030, a local octree-based structure O_k is obtained for said lencompassing cube (C) from said octree information data SOID. In step 1040, one determines if a leaf cube of said local octree-based structure O_k includes a point of the point cloud to be decoded from the second leaf node information data SLID, and in that case, adds a point to the leaf cube of said local octree-based structure O_k.

On **Fig. 1-10**, the modules are functional units, which may or not be in relation with distinguishable physical units. For example, these modules or some of them may be brought together in a unique component or circuit, or contribute to functionalities of a software. *A contrario*, some modules may potentially be composed of separate physical entities. The apparatus which are compatible with the present principles are implemented using either pure hardware, for example using dedicated hardware such ASIC or FPGA or VLSI, respectively « Application Specific Integrated Circuit », « Field-Programmable Gate Array », « Very Large Scale Integration », or from several integrated electronic components embedded in a device or from a blend of hardware and software components.

Fig. 11 represents an exemplary architecture of a device 1100 which may be configured to implement a method described in relation with **Fig. 1-10**.

Device 1100 comprises following elements that are linked together by a data and address bus 1101:

- a microprocessor 1102 (or CPU), which is, for example, a DSP (or Digital Signal Processor);
- a ROM (or Read Only Memory) 1103;
- a RAM (or Random Access Memory) 1104;
- an I/O interface 1105 for reception of data to transmit, from an application; and
- a battery 1106.

In accordance with an example, the battery 1106 is external to the device. In each of mentioned memory, the word « register » used in the specification can correspond to area of small capacity (some bits) or to very large area (e.g. a whole program or large amount of received or decoded data). The ROM 1103 comprises at least a program and parameters. The ROM 1103 may store algorithms and instructions to perform techniques in accordance with present principles. When switched on, the CPU 1102 uploads the program in the RAM and executes the corresponding instructions.

RAM 1104 comprises, in a register, the program executed by the CPU 1102 and uploaded after switch on of the device 1100, input data in a register, intermediate data in different states of the method in a register, and other variables used for the execution of the method in a register.

The implementations described herein may be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal. Even if only discussed in the context of a single form of implementation (for example, discussed only as a method or a device), the implementation of features discussed may also be implemented in other forms (for example a program). An apparatus may be implemented in, for example, appropriate hardware, software, and firmware. The methods may be implemented in, for example, an apparatus such as, for example, a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants ("PDAs"), and other devices that facilitate communication of information between end-users.

In accordance with an example of encoding or an encoder, the point cloud IPC is obtained from a source. For example, the source belongs to a set comprising:

- a local memory (1103 or 1104), e.g. a video memory or a RAM (or Random Access Memory), a flash memory, a ROM (or Read Only Memory), a hard disk ;
- a storage interface (1105), e.g. an interface with a mass storage, a RAM, a flash memory, a ROM, an optical disc or a magnetic support;
- a communication interface (1105), e.g. a wireline interface (for example a bus interface, a wide area network interface, a local area network interface) or a wireless interface (such as a IEEE 802.11 interface or a Bluetooth® interface); and

- an image capturing circuit (e.g. a sensor such as, for example, a CCD (or Charge-Coupled Device) or CMOS (or Complementary Metal-Oxide-Semiconductor)).

In accordance with an example of the decoding or a decoder, the decoded point cloud is sent to a destination; specifically, the destination belongs to a set comprising:

- a local memory (1103 or 1104), e.g. a video memory or a RAM, a flash memory, a hard disk ;
- a storage interface (1105), e.g. an interface with a mass storage, a RAM, a flash memory, a ROM, an optical disc or a magnetic support;
- a communication interface (1105), e.g. a wireline interface (for example a bus interface (e.g. USB (or Universal Serial Bus)), a wide area network interface, a local area network interface, a HDMI (High Definition Multimedia Interface) interface) or a wireless interface (such as a IEEE 802.11 interface, WiFi ® or a Bluetooth ® interface);
- a rendering device; and
- a display.

In accordance with examples of encoding or encoder, the bitstream F1 is sent to a destination. As an example, the bitstream F1 is stored in a local or remote memory, e.g. a video memory (1104) or a RAM (1104), a hard disk (1103). In a variant, the bitstream F1 is sent to a storage interface (1105), e.g. an interface with a mass storage, a flash memory, ROM, an optical disc or a magnetic support and/or transmitted over a communication interface (1105), e.g. an interface to a point to point link, a communication bus, a point to multipoint link or a broadcast network.

In accordance with examples of decoding or decoder, the bitstream F1 is obtained from a source. Exemplarily, a bitstream is read from a local memory, e.g. a video memory (1104), a RAM (1104), a ROM (1103), a flash memory (1103) or a hard disk (1103). In a variant, the bitstream is received from a storage interface (1105), e.g. an interface with a mass storage, a RAM, a ROM, a flash memory, an optical disc or a magnetic support and/or received from a communication interface (1105), e.g. an interface to a point to point link, a bus, a point to multipoint link or a broadcast network.

In accordance with examples, device 1100 being configured to implement an encoding method described in relation with **Fig. 1-9,** belongs to a set comprising:

- a mobile device ;
- a smartphone or a TV set with 3D capture capability
- a communication device ;
- a game device ;
- a tablet (or tablet computer) ;
- a laptop ;
- a still imagecamera;
- a video camera ;
- an encoding chip;
- a still image server ; and
- a video server (e.g. a broadcast server, a video-on-demand server or a web server).

In accordance with examples, device 1100 being configured to implement a decoding method described in relation with **Fig. 10,** belongs to a set comprising:

- a mobile device ;
- a Head Mounted Display (HMD)
- (mixed reality) smartglasses
- an holographic device
- a communication device ;
- a game device ;
- a set top box;
- a TV set;
- a tablet (or tablet computer) ;
- a laptop ;
- a display
- a sterescopic display and

- a decoding chip.

According to an example of the present principles, illustrated in **Fig. 12**, in a transmission context between two remote devices A and B over a communication network NET, the device A comprises a processor in relation with memory RAM and ROM which are configured to implement a method for encoding a colored point cloud as described in relation with the **Figs. 1-9** and the device B comprises a processor in relation with memory RAM and ROM which are configured to implement a method for decoding as described in relation with **Fig. 10**.

In accordance with an example, the network is a broadcast network, adapted to broadcast encoded colored point clouds from device A to decoding devices including the device B.

A signal, intended to be transmitted by the device A, carries the bitstream F1.

This signal may thus carry the first octree information data FOID, the first leaf node information data FLID, the second octree information data SOID and/or the second leaf node information data SLID.

Fig. 13 shows an example of the syntax of such a signal when the data are transmitted over a packet-based transmission protocol. Each transmitted packet P comprises a header H and a payload PAYLOAD.

According to embodiments, the payload PAYLOAD may comprise at least one of the following elements:

- the first octree information data FOID representative of the octree-based structure IO;

- the first leaf node information data FLID indicating if a local octree-based structure O_k has been determined for a LOU_k associated with a leaf node of the octree-based structure IO;

- the second octree information data SOID representative of a local octree-based structure O_k;

- the second leaf node information data SLID indicating if a leaf cube of a local octree-based structure O_k includes a point representative of a part of the point cloud IPC.

Implementations of the various processes and features described herein may be embodied in a variety of different equipment or applications. Examples of such equipment include an encoder, a decoder, a post-processor processing output from a decoder, a pre-processor providing input to an encoder, a video coder, a video decoder, a video codec, a web server, a set-top box, a laptop, a personal computer, a cell phone, a PDA, a HMD, smart glasses, and any other device for processing an image or a video or other communication devices. As should be clear, the equipment may be mobile and even installed in a mobile vehicle.

Additionally, the methods may be implemented by instructions being performed by a processor, and such instructions (and/or data values produced by an implementation) may be stored on a computer readable storage medium. A computer readable storage medium can take the form of a computer readable program product embodied in one or more computer readable medium(s) and having computer readable program code embodied thereon that is executable by a computer. A computer readable storage medium as used herein is considered a non-transitory storage medium given the inherent capability to store the information therein as well as the inherent capability to provide retrieval of the information therefrom. A computer readable storage medium can be, for example, but is not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. It is to be appreciated that the following, while providing more specific examples of computer readable storage mediums to which the present principles can be applied, is merely an illustrative and not exhaustive listing as is readily appreciated by one of ordinary skill in the art: a portable computer diskette; a hard disk; a read-only memory (ROM); an erasable programmable read-only memory (EPROM or Flash memory); a portable compact disc read-only memory (CD-ROM); an optical storage device; a magnetic storage device; or any suitable combination of the foregoing.

The instructions may form an application program tangibly embodied on a processor-readable medium.

Instructions may be, for example, in hardware, firmware, software, or a combination. Instructions may be found in, for example, an operating system, a separate application, or a combination of the two. A processor may be characterized, therefore, as, for example, both a device configured to carry out a process and a device that includes a processor-readable medium (such as a storage device) having instructions for carrying out a process. Further, a processor-readable medium may store, in addition to or in lieu of instructions, data values produced by an implementation.

As will be evident to one of skill in the art, implementations may produce a variety of signals formatted to carry information that may be, for example, stored or transmitted. The information may include, for example, instructions for performing a method, or data produced by one of the described implementations. For example, a signal may be formatted to carry as data the rules for writing or reading the syntax of a described example of the present principles, or to carry as data the actual syntax-values written by a described example of the present principles. Such a signal may be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting may include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The information that the signal carries may be, for example, analog or digital information. The signal may be transmitted over a variety of different wired or wireless links, as is known. The signal may be stored on a processor-readable medium.

A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made. For example, elements of different implementations may be combined, supplemented, modified, or removed to produce other implementations. Additionally, one of ordinary skill will understand that other structures and processes may be substituted for those disclosed and the resulting implementations will perform at least substantially the same function(s), in at least substantially the same way(s), to achieve at least substantially the same result(s) as the implementations disclosed. Accordingly, these and other implementations are contemplated by this application.

CLAIMS

1. A method comprising:

- determining (120) if points $(P_{k,or})$ of a point cloud included in a leaf cube associated with a leaf node of an octree-based structure (IO) representing the geometry of the point cloud are approximated by at least one candidate point $(P_{k,n})$ of a candidate octree-based structure ($O_{k,n}$), as a function of a bit-rate ($R_{k,n}$) for encoding said candidate octree-based structure ($O_{k,n}$) and a distortion ($D_{k,n}$) obtained from spatial distances between, on one hand, said points $(P_{k,or})$ of the point cloud and on the other hand, said candidate points $(P_{k,n})$, a leaf cube associated with a leaf node of said candidate octree-based structure including at most a single candidate point located in its center;

- encoding (130) a first leaf node information data (FLID) indicating if a candidate octree-based structure has been determined for approximating said points $(P_{k,or})$ of the point cloud; and

- if said points $(P_{k,or})$ of the point cloud are approximated by at least one candidate point $(P_{k,n})$, encoding said candidate octree-based structure.

2. A device comprising means for:

- determining if points $(P_{k,or})$ of a point cloud included in a leaf cube associated with a leaf node of an octree-based structure (IO) representing the geometry of the point cloud are approximated by at least one candidate point $(P_{k,n})$ of a candidate octree-based structure ($O_{k,n}$), as a function of a bit-rate ($R_{k,n}$) for encoding said candidate octree-based structure ($O_{k,n}$) and a distortion ($D_{k,n}$) obtained from spatial distances between, on one hand, said points $(P_{k,or})$ of the point cloud and on the other hand, said candidate points $(P_{k,n})$, a leaf cube associated with a leaf node of said candidate octree-based structure including at most a single candidate point located in its center;

- encoding a first leaf node information data (FLID) indicating if a candidate octree-based structure has been determined for approximating said points $(P_{k,or})$ of the point cloud; and

- if said points ($P_{k,or}$) of the point cloud are approximated by at least one candidate point ($P_{k,n}$), encoding said candidate octree-based structure.

3. The method of claim 1 or the device of claim 2, wherein encoding said candidate octree-based structure comprises:

- encoding (140), an octree information data (SOID) representative of said candidate octree-based structure; and

- encoding (150) a second leaf node information data (SLID) indicating if a leaf cube associated with a leaf node of said candidate octree-based structure (O_k) includes a single candidate point located in its center.

4. The method of one of claims 1 or 3 or the device of one of claims 2-3, wherein the method further obtaining or the processor of the device being further configured to obtain said octree-based structure (IO) by splitting recursively a cube encompassing the point cloud until the leaf cubes (LOU_k), associated with the leaf nodes of said octree-based structure (IO), reach down an expected size.

5. The method of one of claims 1 or 3-4 or the device of one of claims 2-4, wherein the method further comprises encoding (110) or the processor of the device is further configured to encode another octree information data (FOID) representative of the octree-based structure (IO).

6. The method of one of claims 1, 3-5 or the device of one of claims 2-5, wherein the distortion ($D_{k,n}$) is obtained from spatial distances between, on one hand, said points ($P_{k,or}$) of the point cloud, and on the other hand, said candidate points ($P_{k,n}$) together with points ($P_{k,IP}$) of an inverse-projected point cloud (IPPC) which are included in said leaf cube associated with a leaf node of said octree-based structure (IO) representing the geometry of a point cloud, said inverse-projected point cloud (IPPC) being obtained by inverse-projecting at least one depth image representative of at least one point of the point cloud.

7. The method or the device of claim 6, wherein the distortion ($D_{k,n}$) is also obtained from neighboring points ($P_{k,NEI}$) which are included in at least one neighboring cube ($LOU_{k,NEI}$) of said leaf cube associated with a leaf node of said octree-based structure (IO) representing the geometry of a point cloud, a neighboring point ($P_{k,NEI}$) being either a point of said inverse-projected point cloud (IPPC) which is included in said at least one neighboring cube ($LOU_{k,NEI}$) or a candidate point located in the center of a leaf cube associated with a leaf node of a candidate octree-based structure previously determined for one of said at least one neighboring cube ($LOU_{k,NEI}$).

8. The method of one of the claims 1-7 or the device of one of claims 2-7, wherein a candidate octree-based structure is obtained by pruning an initial octree-based structure at a given level (le).

9. A method comprising:

- if a first leaf node information data (FLID) indicates that a local octree-based structure has been determined for approximating points ($P_{k,or}$) of a point cloud included in a leaf cube associated with a leaf node of an octree-based structure (IO) representing the geometry of said point cloud,

- obtaining (1030) a local octree-based structure (O_k) for said leaf cube (LOU_k) from an octree information data (SOID);

- if a second leaf node information data (SLID) indicates that a leaf cube associated with a leaf node of said local octree-based structure (O_k) includes a single point, adding a single point in the center of said leaf cube.

10. The method of claim 9, wherein the method further comprises:

- obtaining (1010) an octree-based structure (IO) from another octree information data (FOID);

- for at least one current leaf node of said octree-based structure (IO),

- if a first leaf node information data (FLID) indicates that a local octree-based structure has been determined for approximating points ($P_{k,or}$) of a point cloud included in a leaf cube associated with said current leaf node,

- obtaining (1030) a local octree-based structure (O_k) for said leaf cube (LOU_k) from an octree information data (SOID);

- if a second leaf node information data (SLID) indicates that a leaf cube associated with a leaf node of said local octree-based structure (O_k) includes a single point, adding a single point in the center of said leaf cube.

11. A device comprising a processor configured to:

- if a first leaf node information data (FLID) indicates that a local octree-based structure has been determined for approximating points ($P_{k,or}$) of a point cloud included in a leaf cube associated with a leaf node of an octree-based structure (IO) representing the geometry of said point cloud,

- obtain (1030) a local octree-based structure (O_k) for said leaf cube (LOU_k) from an octree information data (SOID);

- if a second leaf node information data (SLID) indicates that a leaf cube associated with a leaf node of said local octree-based structure (O_k) includes a single point, add a single point in the center of said leaf cube.

12. The device of claim 11, wherein the processor is further configured to:

- obtain an octree-based structure (IO) from another octree information data (FOID);

- for at least one current leaf node of said octree-based structure (IO),

- if a first leaf node information data (FLID) indicates that a local octree-based structure has been determined for approximating points ($P_{k,or}$) of a point cloud included in a leaf cube associated with said current leaf node,

- obtain a local octree-based structure (O_k) for said leaf cube (LOU_k) from an octree information data (SOID);

- if a second leaf node information data (SLID) indicates that a leaf cube associated with a leaf node of said local octree-based structure (O_k) includes a single point, add a single point in the center of said leaf cube.

13. A signal carrying:

- a first octree information data (FOID) representative of an octree-based structure (IO) representing the geometry of a point cloud;

- for at least one current leaf cube (LOU_k) associated with a current leaf node of said octree-based structure (IO),

- a first leaf node information data (FLID) indicating if a local octree-based structure has been determined for said current leaf cube;

- if said first leaf node information data (FLID) indicates that a local octree-based structure has been determined

- a second octree information data (SOID) representative of said determined local octree-based structure (O_k); and

- at least one second leaf node information data (SLID) indicating if a leaf cube associated with a leaf node of said local octree-based structure (O_k) includes a single point in its center.

14. A non-transitory computer readable medium comprising instructions which, when executed by a computer, cause the computer to carry out the method of claim 1 or 9.

15. A computer program product comprising instructions which, when the program is executed by a computer, cause the computer to carry out the steps of the method of claim 1 or 9.

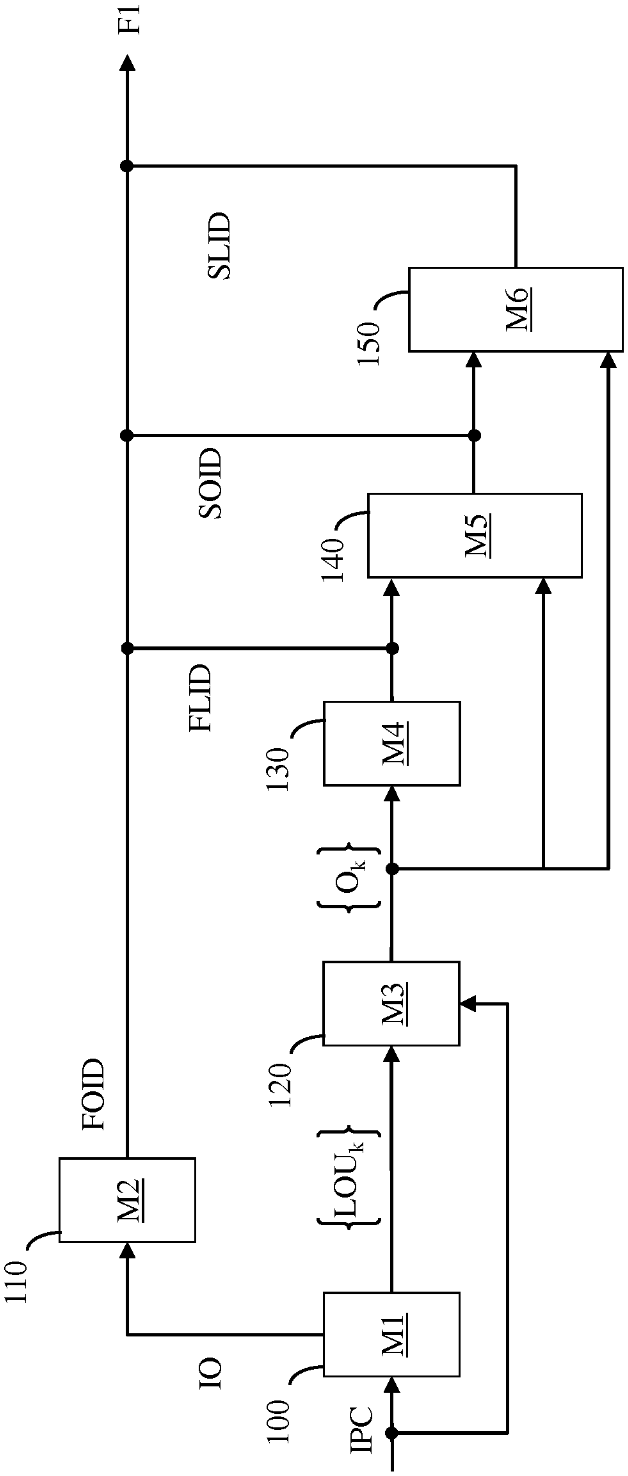

F1
SLID
SOID
FLID
FOID
M1
M2
M3
M4
M5
M6
100
110
120
130
140
150
IPC
IO
{LOU_k}
{O_k}

Fig. 1

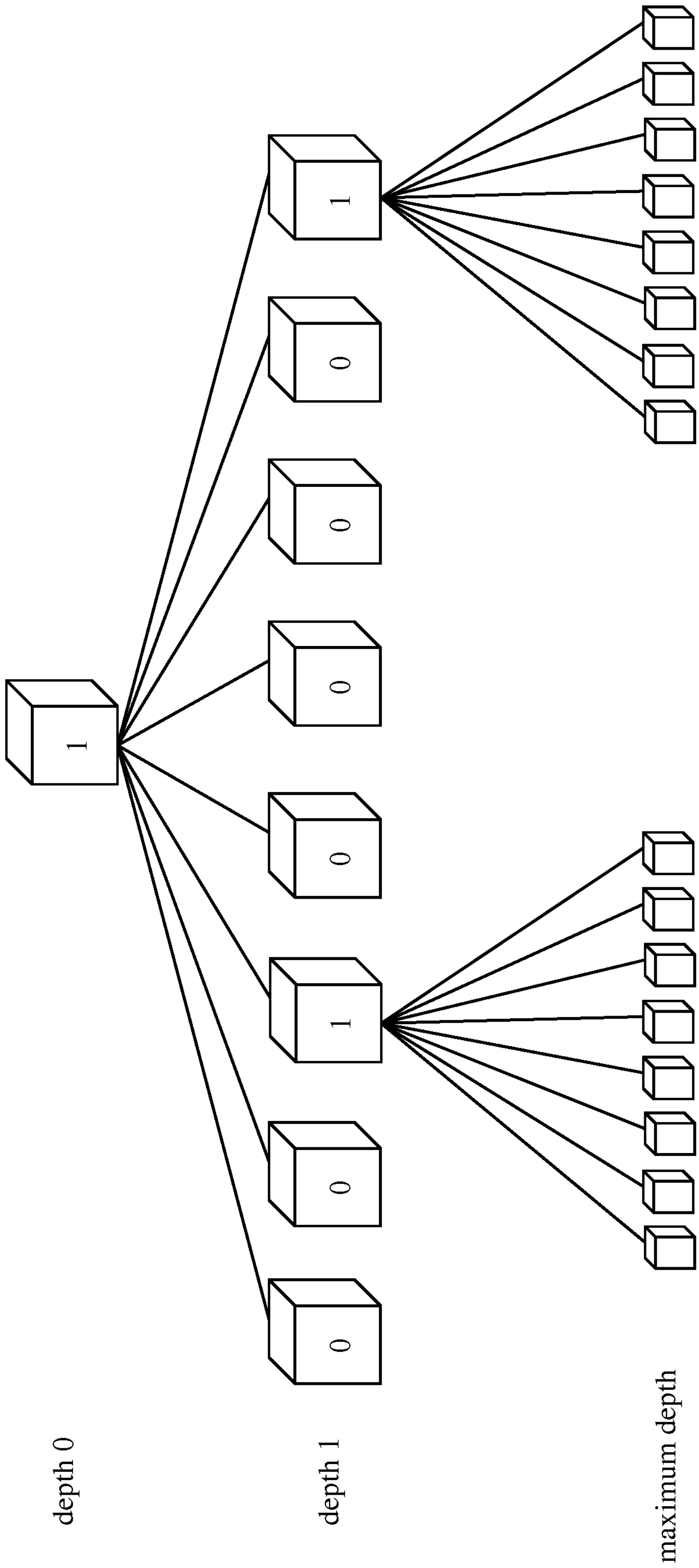


Fig. 2

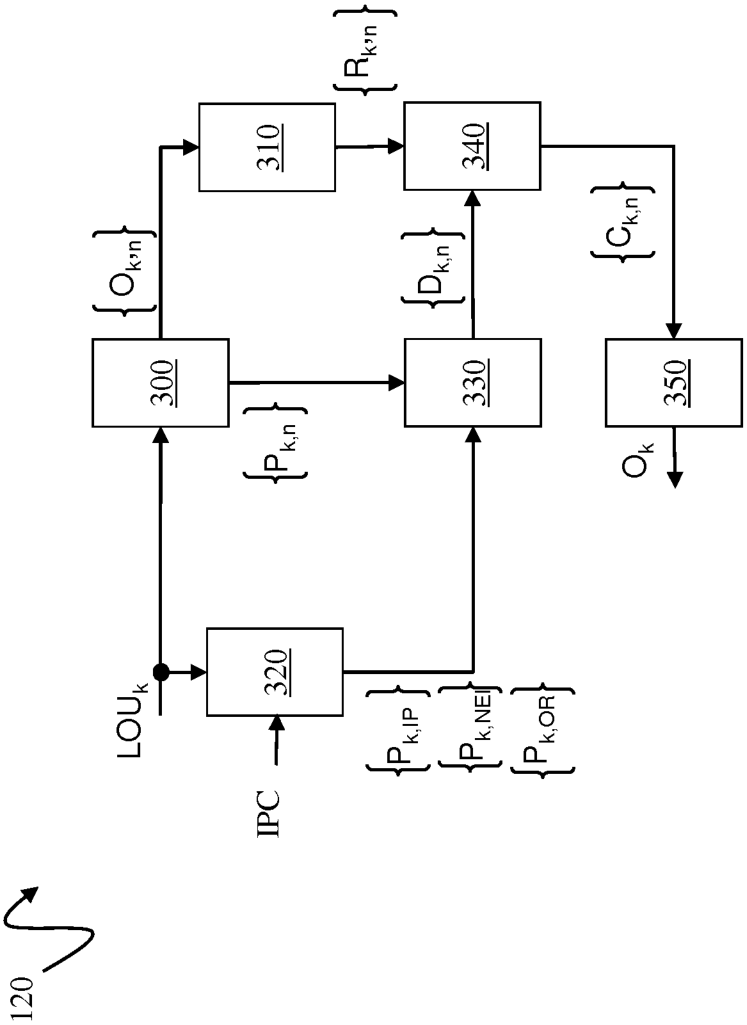

120
IPC
LOUk
320
300
310
340
330
350
{Ok,n}
{Pk,n}
{Rk,n}
{Dk,n}
{Ck,n}
Ok
{Pk,IP}
{Pk,NEI}
{Pk,OR}

Fig. 3

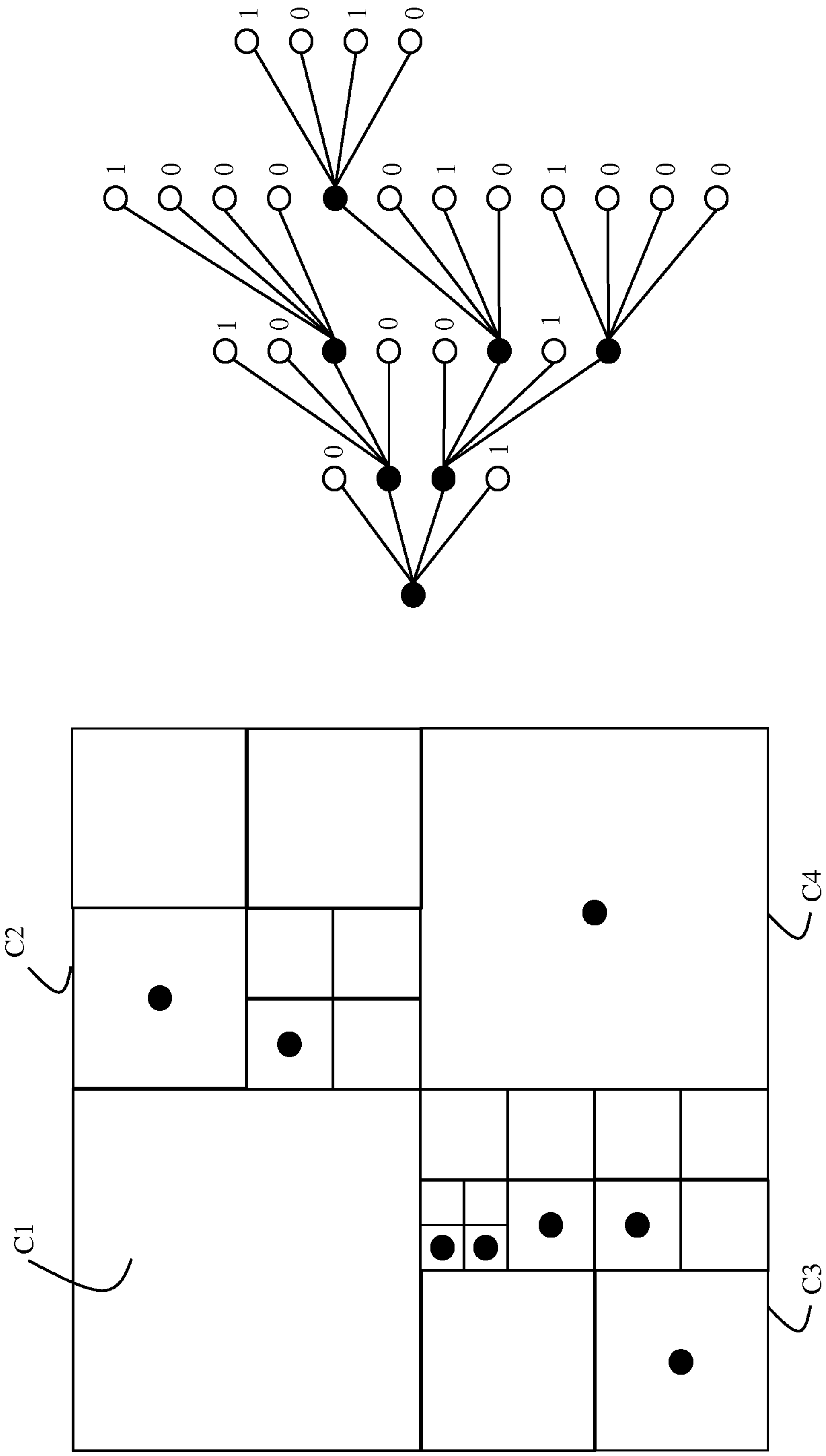


Fig. 4

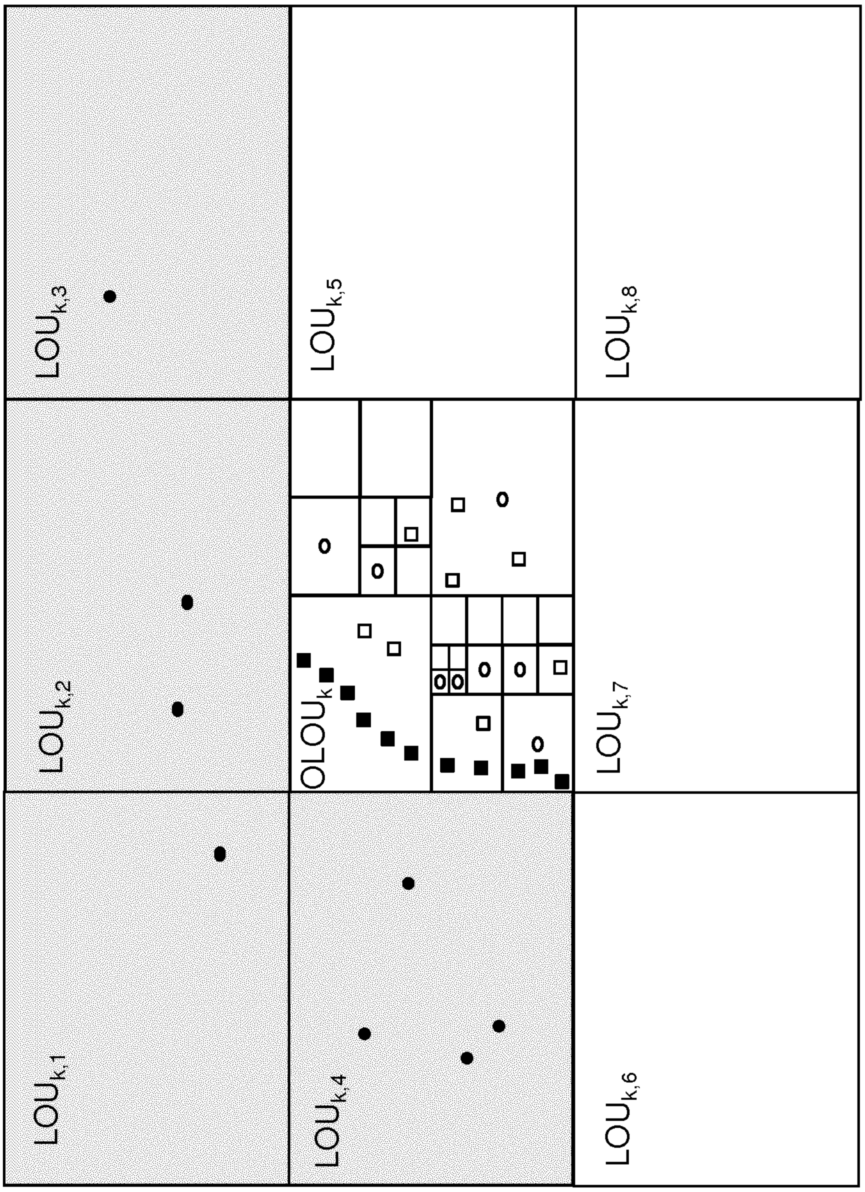


Fig. 5

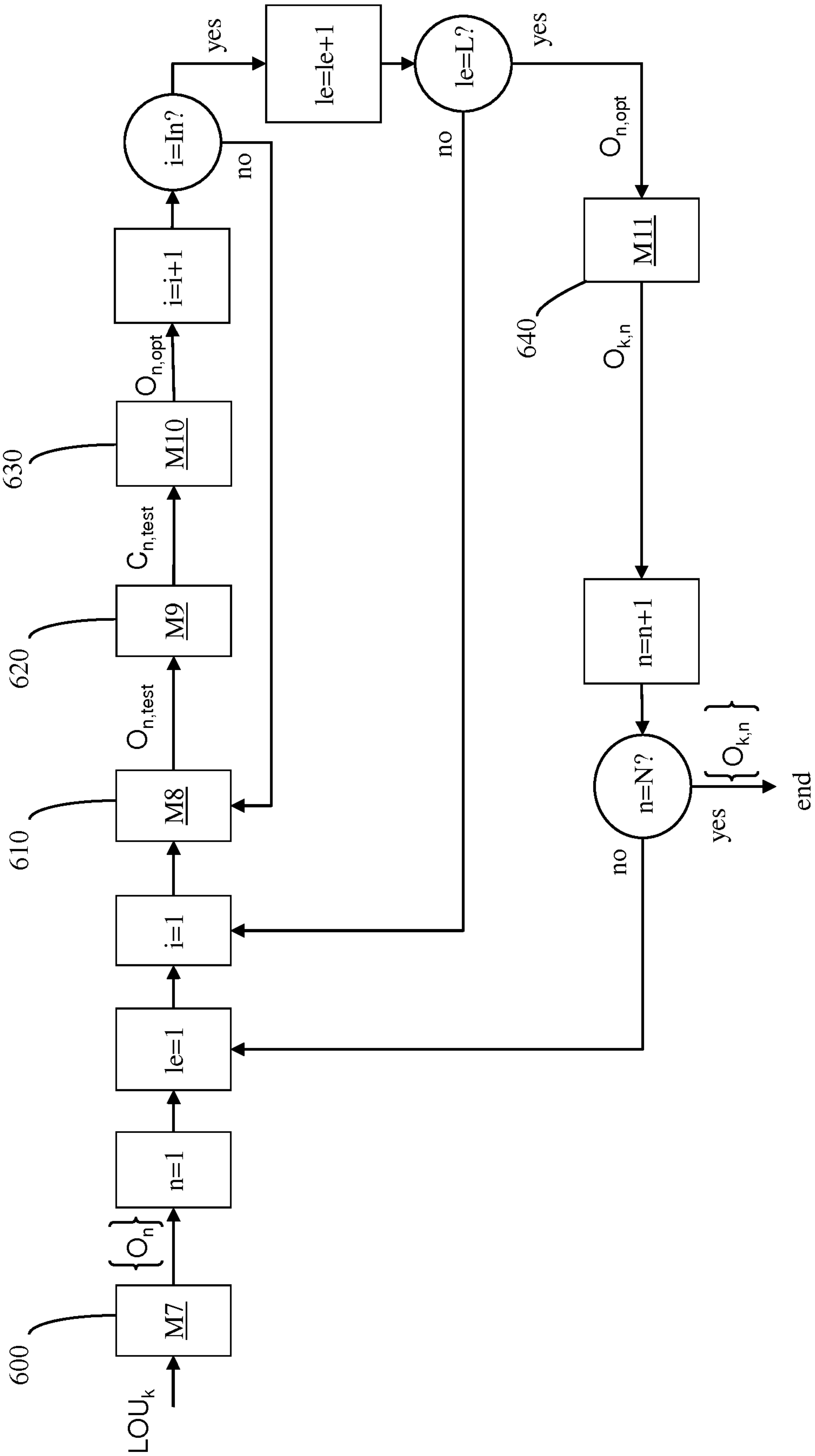

LOU$_k$
600
M7
{O$_n$}
n=1
le=1
i=1
610
M8
O$_{n,test}$
620
M9
C$_{n,test}$
630
M10
O$_{n,opt}$
i=i+1
i=In?
yes
no
le=le+1
le=L?
yes
no
O$_{n,opt}$
640
M11
O$_{k,n}$
n=n+1
n=N?
no
yes
{O$_{k,n}$}
end

Fig. 6

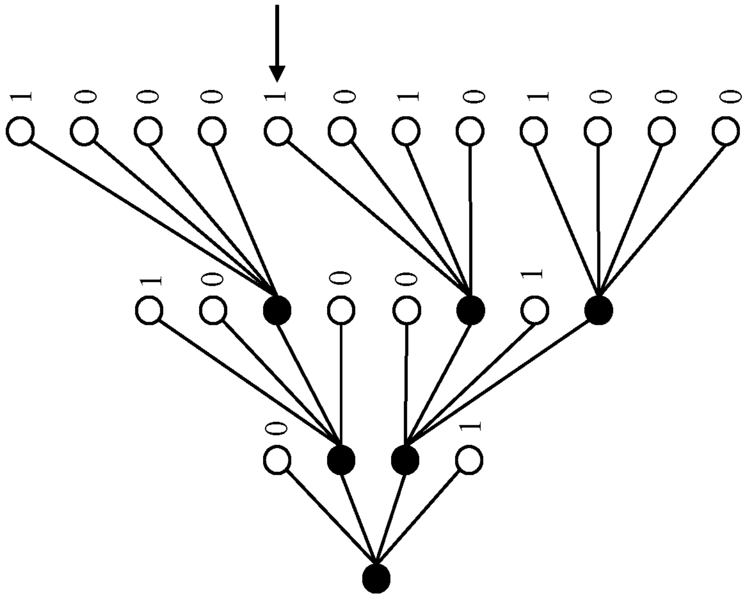


Fig. 7b

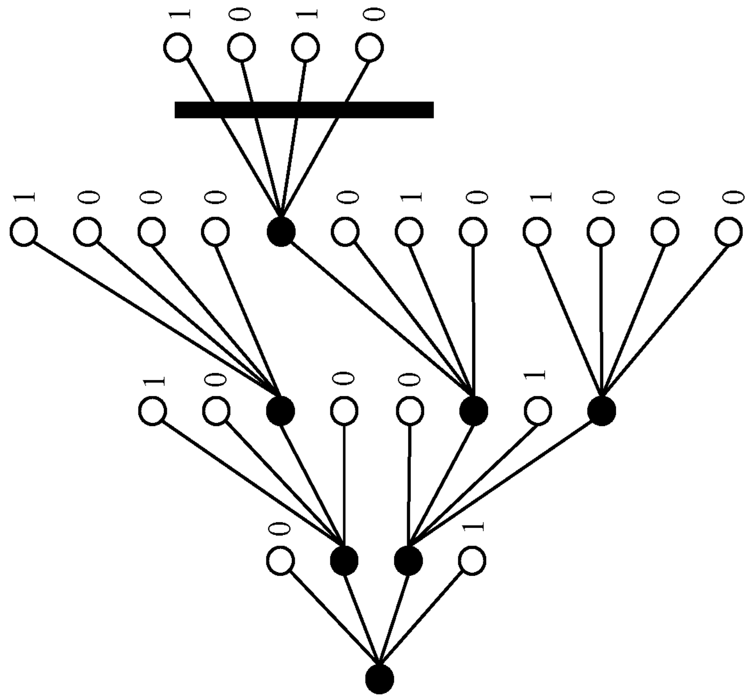


Fig. 7a

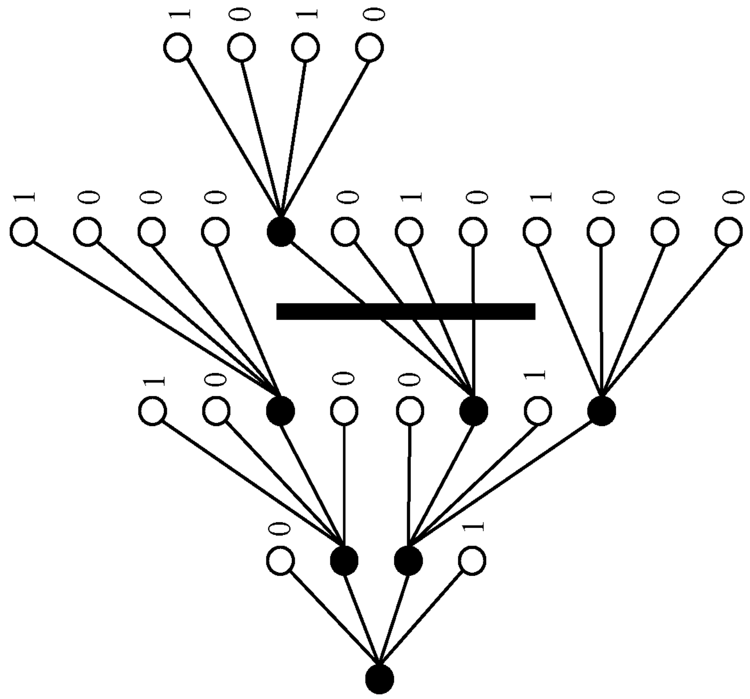


Fig. 8a

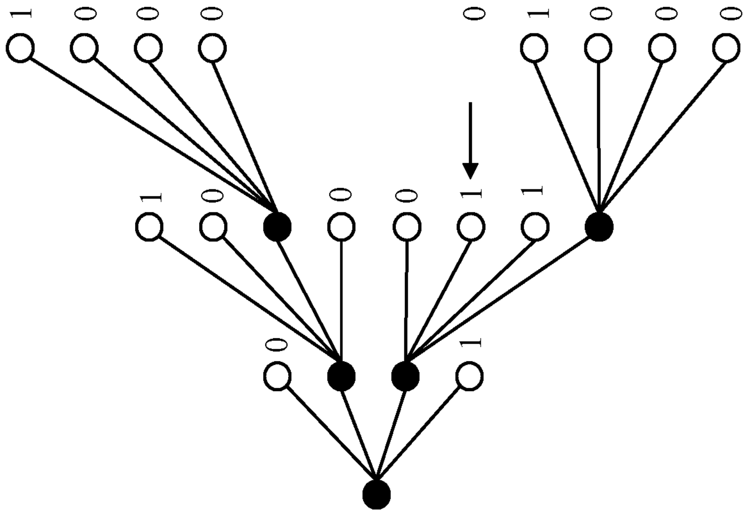


Fig. 8b

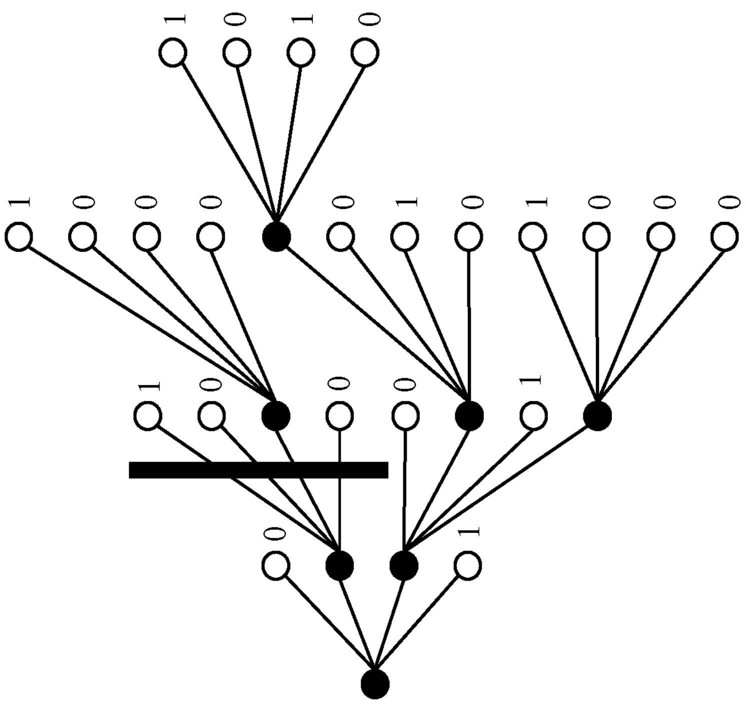


Fig. 9a

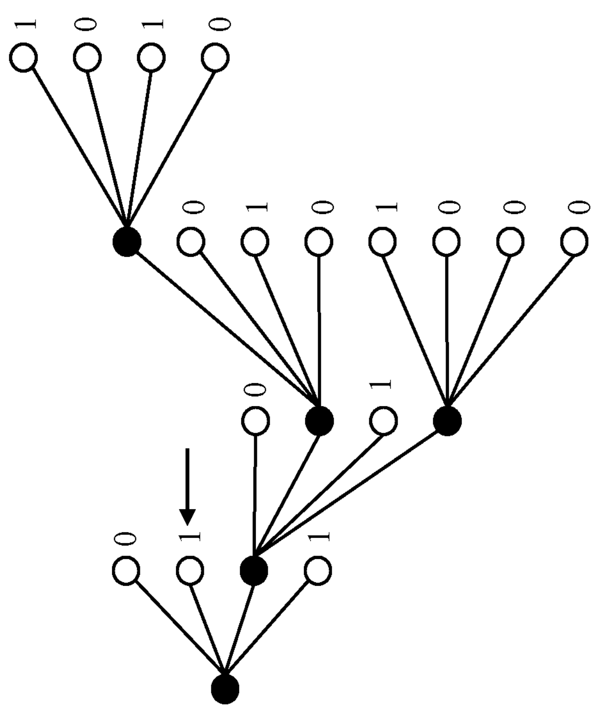
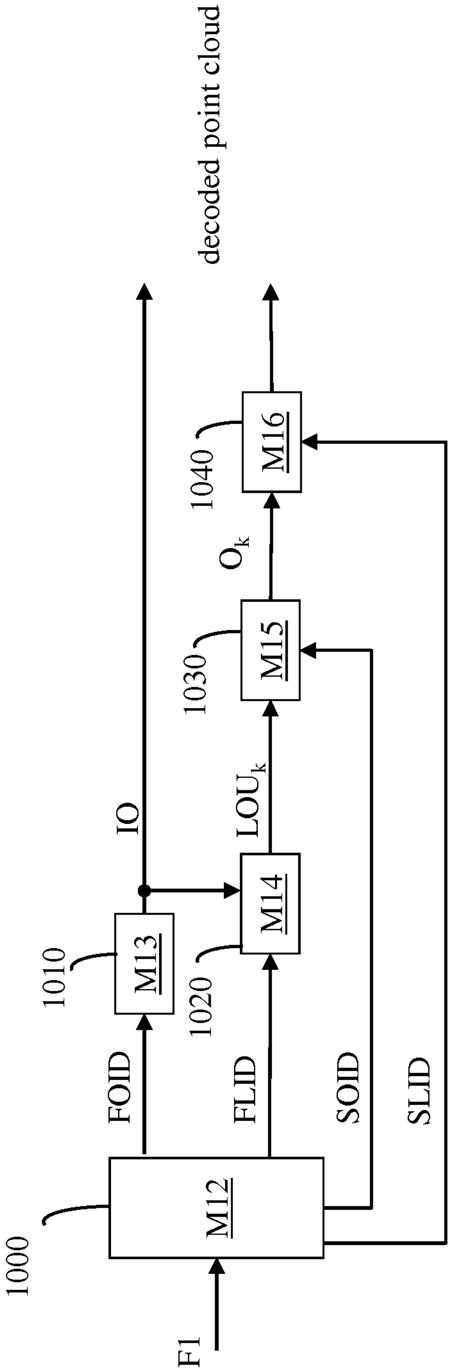


Fig. 9b



decoded point cloud
IO
FOID
FLID
SOID
SLID
LOU_k
O_k
M12
M13
M14
M15
M16
1000
1010
1020
1030
1040
F1

Fig. 10

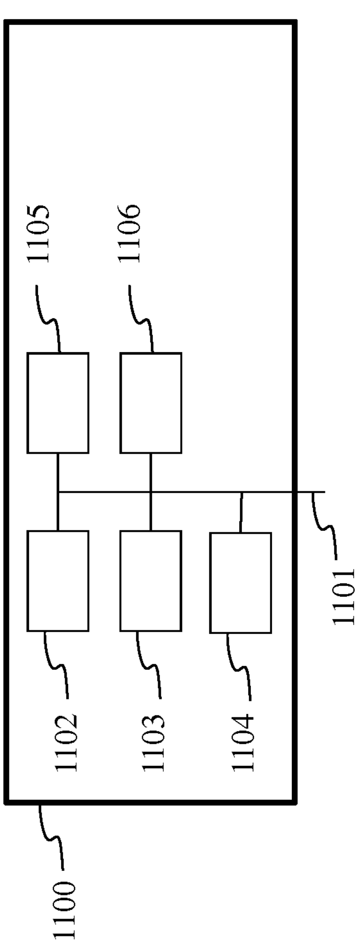


Fig. 11

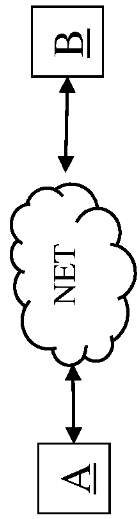


Fig. 12

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2018/066923

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/96 G06T9/40 H04N19/147 H04N19/597 G06T17/00
G06T17/10 H04N19/20
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 431 919 A1 (SAMSUNG ELECTRONICS CO LTD [KR]; SEOUL NAT UNIV IND FOUNDATION [KR]) 23 June 2004 (2004-06-23) abstract; figures 3,5-7, 16-23 paragraphs [0001], [0010] - [0019], [0039] - [0073] -----	1-15
X	BIRENDRA KATHARIYA (UMKC) ET AL: "Dynamic Geometry Compression with Binary Tree Based Prediction and Residual Coding", 118. MPEG MEETING; 3-4-2017 - 7-4-2017; HOBART; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11),, no. m40607, 31 March 2017 (2017-03-31), XP030068952,	9,11, 13-15
A	paragraphs [0001] - [0003]; figure 5 -----	1-8,10, 12

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 July 2018	23/07/2018

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Cakiroglu Garton, S

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2018/066923

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	RICARDO L DE QUEIROZ ET AL: "Motion-Compensated Compression of Dynamic Voxelized Point Clouds", MPEG 115TH MEETING GENEVA, 25 May 2016 (2016-05-25), XP055395321, abstract paragraphs [00IV], [000V] -----	9,11, 13-15
A	KALAIAH ET AL.: "Statistical Geometry Representation for Efficient Transmission and Rendering", ACM, 2 PENN PLAZA, SUITE 701 - NEW YORK USA, vol. 24, no. 2, 1 April 2005 (2005-04-01), pages 348-373, XP040018890, the whole document -----	1-13

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2018/066923

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1431919 A1	23-06-2004	CN 1549206 A	24-11-2004
		EP 1431919 A1	23-06-2004
		JP 4166679 B2	15-10-2008
		JP 2004185628 A	02-07-2004